

My opinionated advise on how to learn vim.

First if you do not type in 10 finger system then you should learn it before.

Tipp10 is great to learn the 10 finger system. Link to Software: <https://www.tipp10.com/en/> For linux you can install via flatpak under <https://flathub.org/apps/details/com.gitlab.tipp10.tipp10>

Plan a few weeks for getting used to it.

These document instructs you to learn vim step by step in an interval of weeks.

Reason: I believe that learning too many things at once, will frustrate you so much that you will give up in learning vim.

From 1. to 3. week is learning the vim mode in your IDE/Text editor of your choice. From 4. week on is optional. You decide if you want to switch to vim itself as your tool.

Week 1: Start with vim bindings in your IDE !

Install the plugin\extension for the vim mode in your editor

Learning and building up the muscle memory for vim bindings is hard at the start. By learning vim bindings in your used development enviroment, your favorite IDE, you can focus only on that. Using vim right away forces you to learn vim plugging, configuration system and vim script at the same time too !

For example visual studio code has the extension called VSCodeVim. Link: <https://marketplace.visualstudio.com/items?itemName=vscodevim.vim:w>

Move left, right, up and down

↑ ← → ↓

Even better: Hard at the start but worth it

k h l j

Go to Insert Mode:

i

Go back to Normal Mode from any other mode like command/visual mode

| esc | or | ctrl + { | or | ctrl c |

Go to visual mode

v

Go to command mode

:

Bare minimum File operations in commmand mode

:q Quit buffer after saving

:q! Quit buffer without saving

:w Save buffer

:wq Save and quit buffer

Copy and paste

y Yank/Copy Text

p Paste d Cut/Delete selected text

Undo and redo in normal mode

u undo ctrl + r redo

Useful base commands in normal mode

- b | move one word back
- w | move one word forward
- e | move to next end of a word
- yy | yank line
- dd | cut line
- yw | copy word
- dw | cut word
- shift + v | select line by line in visual mode

Week 2

Moving left, right, up and down

Remember these keys for left, right, up and down

k h l j

If you still use these ↑ ← → ↓

Stop and get used to

k h l j

Finally !

Search and replace in the vim way

Link how it is done in vim manner https://vim.fandom.com/wiki/Search_and_replace

For most editor it works the same way like in vim. Go into command mode with ":" then type / for search for s/ for replace.

- gg go to end of file
- G got to start of file
- Line jumping Example: :120 go to line 120
- Start learning via vimtutor

open vim and type :vimtutor Do it again here and there

Learn configuring keybindings in your editor

Google it for editor plugin !

3. Week More find grained key binding

Vertical speed

- ctrl + u go a half page up, many lines.
- ctrl + d go a half page down, many lines.
- Set option to relative numbers line. Then use and j and k with numbers to move to specific lien fast.
 - Example: 10 j to move 10 lines down

horizontal speed

- t {some_key} goes right before {some_key}. Example tf goes right before next f.
- f {some_key} go to next {some_key}. Example ff goes to next f.
 - n go to next {some_key}
 - , got previous {some_key}
- shift + ^ go to start of line
- shift + \$ go to end of line
- shift + i go to end of line and go to insert mode
- shift + a go to start of line and go to insert mode
- shift + d cut right of cursor to end line
- shift + y yank right of cursor to end line
- shift + c cut right of cursor to end line and go to insert mode

Text manipulation speed

- shift + p paste before cursor/above line.
- shift + o insert new line above
- x Remove letter under cursor
- r Change 1 letter
- s Remove letter under cursor and go to insert mode.j
- cc cut line and got to insert mode
- cw cut word and got to insert mode

Last Week 4 if you want to stick with your favorite IDE/Editor.

More vim concepts

- Learn about the numerous registers/clipboards in vim. <https://www.briantorti.com/vim-registers/>
 - Learn searching/replacing text via / and s/
-

Week 4: you want to use vim/nvim as ide

Heads up: new learning curve is coming. You need plugins and configuration to turn vim into an IDE.

start customizing vim

- Learn to configure the .vimrc file.
- Install a plugin manager, vim-plug for example.
- learn to install at least a plugin and configure it.
- Learn to target a specific branch and version/tag of plugin on github for plugin
 - Reason: Without it you get the last commit. This can lead to more bugs.
- Learn use the help system via :h, :help and helpgrep.
 - Helpgrep searches for word throughout the document system.
 - Read: Only :h or :help without any word brings you to the main page of the vim doc. Read it !
- Use vim as your ad hoc text editor for normal text and config files.

Decide if you want to continue using vim or use nvim instead.

Link to nvim: <https://neovim.io/>

Vim and nvim can be good IDE. Many things learned from vim can be transferred to nvim. It is okay to stick to vim a bit longer.

However I recommend to switch to nvim later or better start using it now.

Reason for nvim out of my view:

- Nvim is configurable via vim script and lua easily.
- Lua is not perfect but is way saner than vimscript and faster
- Nvim is known to have the innovating features. Vim is adapting theme later.
- Nvim is developed by community and not by only one person like in vim.

5. Week Get vim and nvim as IDE right away

Start with a distribution for vim/nvim

A distribution is a folder with a strong default configuration for vim/nvim. It sets up a plugin packer, has many packages selected and configured.

This provides you with IDE from the start. The can further customized/extended to you liking. It also teaches your what plugins are well established and can be viewed as example for complex configuration vim.

As moment of writing I am at the stage of using such distribution.

- Astrovim/Lunarvim for nvim
- Spacevim for vim or nvim

Consider using vim/nvim as window program instead of a terminal

- Gvim for vim
- Neovide for nvim

Or use tmux or a terminal simulator with multiple tabs.

Many weeks later after 4

Decide if you want to configure vim/nvim from scratch for max control

I am not sure if I do this myself.

- Get confident with scripting language
 - vimscript for vim. Have fun
 - Lua for nvim. Write your config then completely in Lua even if vimscript can also be used.
- Evaluate the which plugins are needed for workflow
- Split up your configuration in several files.

Learn resources

- Video Playlist on youtube: https://www.youtube.com/watch?v=X6AR2RMB5tE&list=PLm323Lc7iSW_wuxqmKx_xxNtjC_hJbQ7R
- Link to vim registers: <https://www.briantorti.com/vim-registers/>
- Link for how to think more in English for vim, help for memorizing key bindings: <https://www.youtube.com/watch?v=2JURtQhBvGs>